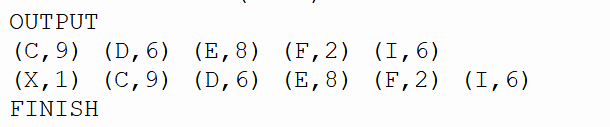


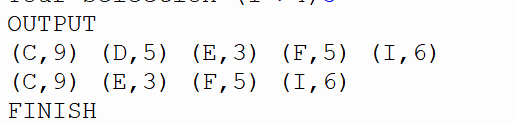
1.1

Check xem thỏa mãn điều kiện không, nếu thõa mãn điều kiện add vào đầu danh sách, chỉnh lại node đầu thành next, node mới thành head



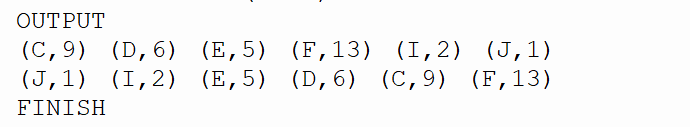
1.2

Hàm sử dụng linklist đầu tiên kiểm tra xem chuỗi có trống không nếu có thì x sẽ là node đầu luôn còn nếu sai thì data.head sẽ bằng data.next và node x sẽ là date.head mới



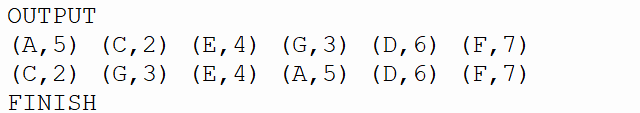
1.3

sử dụng con trỏ crr để kiểm tra liên kết nào có price bằng 5 nếu tìm thấy price bằng 5 node ở đó sẽ bị xóa bỏ và các node khác sẽ liên kết lại với nhau nếu crr bằng NONE( đã chạy hết linklist) return



1.4

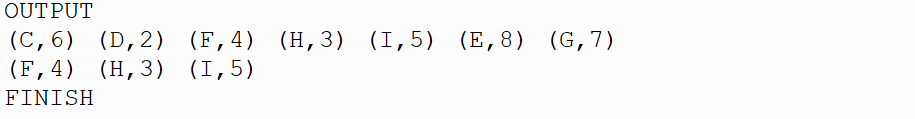
Đoạn mã định nghĩa một phương thức f4 trong một lớp, được thiết kế để sắp xếp một danh sách liên kết theo giá trị Price tăng dần. Phương thức sử dụng thuật toán sắp xếp nổi bọt (bubble sort) để so sánh và hoán đổi các nút trong danh sách liên kết. Quá trình này tiếp tục cho đến khi toàn bộ danh sách được sắp xếp theo thứ tự mong muốn. Nếu danh sách liên kết ban đầu là None, phương thức kết thúc ngay lập tức mà không thực hiện bất kỳ thao tác nào.



2.1

Nếu tên bắt đầu bằng "B" hoặc giá lớn hơn 100, phương thức trả về ngay lập tức mà không thực hiện chèn nút,Tạo một đối tượng Node mới, chứa một đối tượng Car với tên và giá được cung cấp.Nếu cây không rỗng, phương thức sẽ duyệt qua cây để tìm vị trí thích hợp để chèn nút mới.

Trong quá trình duyệt, nếu tìm thấy một nút có Price bằng với giá của nút mới, phương thức sẽ trả về ngay lập tức, không chèn nút mới (điều này ngăn chặn sự trùng lặp).

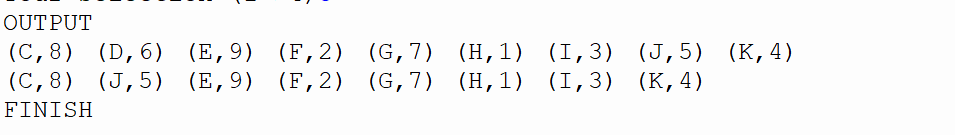


Q2\_2:

Duyệt cây theo thứ tự trước (pre-order), bắt đầu từ nút p, nếu nút p là none (điều kiện dừng), phương thức kết thúc, Kiểm tra nút p, bằng cách sử dụng phương thức check. Nếu nút thỏa mãn điều kiện trong check(price trong khoảng 3-5), thực hiện phương thức visit đối với nút đó.

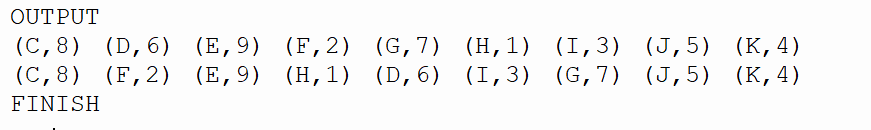
visit:

Nếu nút p là None, phương thức kết thúc, nếu thỏa mãn In dữ liệu của nút p (được giả định là thông tin về một đối tượng Car).



2.3

Mã Python này triển khai lớp Q2\_3 để xóa một nút trong cây nhị phân. Nó sử dụng duyệt cây theo chiều rộng để tìm nút cần xóa dựa trên điều kiện check. Hàm deleteByCopyingLeft thực hiện xóa bằng cách sao chép giá trị từ nút bên trái nhất của con phải. Hàm chính f3 gọi các phương thức trên để thực hiện xóa



2.4

Đoạn mã thực hiện duyệt cây nhị phân theo chiều rộng và thực hiện phép xoay cây phải (right rotation) khi gặp một điều kiện cụ thể trên một nút. Cụ thể:

Duyệt cây theo chiều rộng và kiểm tra mỗi nút.

Nếu một nút thỏa mãn điều kiện cụ thể (check(p.data.Price) và flag), lưu trữ nút đó và tăng giá trị count.

Nếu đã tìm thấy nút đáp ứng điều kiện, thực hiện phép xoay cây phải tại nút đó.

Mục đích chính có vẻ liên quan đến việc cân bằng cây nhị phân hoặc thực hiện một tối ưu hóa nào đó khi gặp điều kiện cụ thể trong quá trình duyệt.